Simuler une population à échantillonner et évaluer les meilleures stratégies pour faire évoluer et compléter un dispositif existant afin de gagner en précision et/ou en puissance de détection
l'exemple de LDSO - TP

LDSO se lance sur DGA12 er DGA11 par la simple commande `ldso`. Les numéros de page du TP font référence à la notice.

L'exécution de LDSO requiert la présence de 5 fichiers paramètres pour l'exécution, dont les noms sont fixes:

- `general`
- `pop1`
- `popfin`
- `files`
- `param`

L'essentiel de la sortie est donnée à l'écran, mais peut être imprimée vers un fichier par la commande `ldso > fichier`. Pour chacun des exemples ci-dessous, les résultats de l'exécution ont été archivés dans des fichiers nommés `result` (sauf pour POP, fichier `work.o.155219`), que vous pourrez utiliser en cas de difficultés d'exécution des exemples donnés.

Le dossier contient 10 dossiers, qui correspondent pour les 8 premiers à des exemples de situations simulées différentes explicités dans la notice, et pour les deux derniers aux données simulées pour la formation (p48 et suivantes):

`EX1 à EX8`

POP (jeu de données complet)

`POPred` (jeu de données réduit)

Le TD s'appuie sur l'exemple rassemblé dans `POPred`, et sa comparaison avec les résultats obtenus dans POP. Il s'agit de simuler une structure de DL complexe, proche de celle utilisée pour la manipulation des logiciels pendant la formation (dossier POP). Les points 1 à 9 peuvent être traités en parallèle pour le dossier `POPred` et le dossier POP.

1) Examiner la structure du fichier d'entrée `general`

lignes 1 à 20

lignes 5017 et suivantes / 10017 et suivantes

Quelles est la position des QTL simulés ? Quels sont leurs effets ? Sont ils soumis à mutation ?

2) Examiner la structure du fichier d'entrée `pop1` : pour faciliter l'exécution en TD, la partie historique décrite dans ce fichier a été réduite à 10 générations à 1000 individus pour commencer, plutôt que 1000 tel que paramétré dans le dossier POP.

lignes 1 à 10

ligne 1016 - chercher l'autre QTL / les autres QTL

ligne 5017 et suivantes / 10017 et suivantes

Quels sont les évènements qui structurent le DL dans la population finale ? (goulots d'étranglement ? Expansion ? Sélection ?)

3) Examiner la structure du fichier d'entrée `popfin`: pour faciliter l'exécution en TD, la partie pedigree de `POPred` a été réduite à 5 générations plutôt que 10 comme paramétré dans le dossier POP.

Quelle est la structure récente du pedigree ? Combien d'individus attendez vous dans les fichiers de sortie ?

4) Lancer l'exécution dans le dossier `POPred`. Pour faciliter l'exécution, il est recommandé de se connecter sur DGA11 (`qlogin -q`) ou d'y lancer les exécutions (`qsub ldso`). L'ouverture des fichiers peut être maintenue sur DGA12.

Examiner la sortie que vous avez produite dans `POPred`, et celle stockée dans le fichier `work.o.155219` pour `POP` (ne pas lancer l'exécution, trop long), vérifier l'exactitude des impressions de la sortie principale (*nombre de générations, nombre et positions des QTL, taille de la population à chaque génération*) en comparant avec les données des fichiers paramètres.

```
 1
 general read
 pop1 read
 QTL position(s)        1001        3808
 DOMINANCE AND EPISTATIC EFFECTS: DONE
 CREATION OF THE FIRST POPULATION
 Sizes population 1         1000         1000         1000         1000         1000         1000
1000         1000         1000         1000         1000          150          150          150
150          150          150          150          150          150          150          150
150          150          150          150          150          150          150          150
150          150          150          150          150          150          150          150
150          150          150          335          520          705          890         1075
1260        1445        1630        1815        2000
 GENERATION           1
 GENERATION           2
 GENERATION           3
 .
 .
 .
 GENERATION          47
 GENERATION          48
 GENERATION          49
 GENERATION          50
 CREATION PHENO          50
 End of historical generations
 POPFIN READ
 GENE           1
 GENE           2
 GENE           3
 GENE           4
 GENE           5
```

Exemple de sortie générale pour `POPred.`

5) Examiner les fichiers de sortie de routine (p42-44):

`map` (num marqueur, chromosome, id_marqueur, position, QTL/no QTL)

Quelles sont les positions pour chaque QTL?

`effects` (une ligne par QTL: effet additif simulé pour chaque allèle (tous à 0 sauf 1 à +a et 1 à -a, a étant donnés dans le fichier `general`; effet de dominance)

Quels sont les allèles à effet pour chaque QTL?

`simped`
`simhaplo`
`simhaploNoQTL`
`simcop`
`simcopNoQTL`
`simperf`

heterozygotes
genotyp_err

Y a-t-il des erreurs de génotypage simulées?

6) Utiliser la commande ci dessous pour générer un fichier ne contenant que les génotypes au QTL des individus (2 lignes par individu).

dossier POPred :

```
gawk '{print $1 " " $2 " " $1002 " " $3810}' simhaplo > simQTL
```

dossier POP :

```
gawk '{print $1 " " $2 " " $1002 " " $3810 " " $6004 " " $6666 " " $7984}' simhaplo > simQTL
```

Utiliser le logiciel de votre choix (par exemple R, script freqall.r) pour estimer les fréquences alléliques et génotypiques pour chaque QTL. *En combinant cette information avec celle issue du fichier* `effects`*, que concluez vous quant à l'effet du QTL pour les 10 premiers individus du fichier? au niveau de la population ?*

6bis) Utiliser la commande ci-dessous pour générer un fichier ne contenant que les origines des allèles au QTL des individus (2 lignes par individu).

dossier POPred :

```
gawk '{print $1 " " $2 " " $1002 " " $3810}' simcop > simcopQTL
```

dossier POP :

```
gawk '{print $1 " " $2 " " $1002 " " $3810 " " $6004 " " $6666 " " $7984}' simcop > simcopQTL
```

Utiliser le logiciel de votre choix (par exemple R, script `copie.r`) pour estimer le nombre d'origines représentées pour chaque QTL : *qu'en concluez vous sur l'effet de la dérive et de la sélection sur la variabilité? Que peut on supposer quant à la consanguinité des animaux?*

Quel est l'effet de simuler beaucoup plus de générations dans le dossier POP sur l'origine du DL? Sur le polymorphisme des SNP et des QTL ?

7) Examiner le fichier files (p 45 à 47), et les fichiers de sortie générés par les options retenues

simld
simldq
simfqall

8) Utiliser la commande ci-dessous pour sélectionner les sorties relatives au DL entre marqueurs pour les marqueurs espacés de moins de 20 cM.

```
gawk '$7<0.22 {print $0}' simld > simldtemp2
```

Avec R (`LD_graph.r`), générer deux graphs représentant la décroissance du DL en fonction de la distance entre les marqueurs dans le jeu de donnée simulé, le premier entre 0 et 20 cM, le deuxième en faisant un zoom entre 0 et 2cM.

Quel est l'effet de simuler beaucoup plus de générations dans le dossier POP sur la structure de DL et son étendue ?

9) Utiliser les commandes ci-dessous pour

compter le nombre de lignes du fichier: une par allèle simulé = 4999SNP * 2 allèles + 2 QTL * 5 allèles pour POPred,

puis récupérer uniquement les fréquences des allèles 1 de chaque marqueur et QTL,

compter le nombre de loci (SNP + QTL) simulés,

récupérer uniquement les fréquences des marqueurs polymorphes (plus les QTL dont l'allèle 1 n'a pas une fréquence égale à 0 ou 1),

et enfin compter le nombre de loci (SNP + QTL pour allèle 1) polymorphes.

```
wc -l simfqall
gawk '$2==1 {print $0}' simfqall > simfqall1
wc -l simfqall1
gawk '$3>0 && $3<1 {print $0}' simfqall1 > simfqall2
wc -l simfqall2
```

Combien de marqueurs (aux QTL près) sont devenus monomorphes pendant la simulation?

Représenter la distribution des fréquences alléliques obtenues (par exemple avec R, freqall.r).

10) Si ce n'est pas déjà fait, répéter les points 1 (SANS lancer l'exécution, trop long) à 9 sur les fichiers du dossier POP.

11) A l'aide de la notice et des scripts fournis ci-dessus, explorer les exemples EX1 à EX8, faites varier les paramètres de simulation.

12) L'ensemble des résultats obtenus est lié aux racines ('seed') de simulations données dans les fichiers param. Vous pouvez générer de nouveaux jeux de données en changeant ces valeurs. Pour une étude qui s'appuierait sur des simulations successives nombreuses, l'idéal est de générer automatiquement des valeurs différentes à chaque nouvelle simulation.

```
2557 3574352 85413 ! seeds for the simulations : genetic map , historical population, final population
```

Note1 :

```
wc -l <fichier>
```

donne le nombre de lignes de <fichier>

Note2 : gawk est utilisé ici pour des syntaxes simples du type

```
gawk '<condition> {print $num_colonne " " $num_colonne}' <fichier_source> > <fichier_sortie>
```

ou

```
gawk '<condition> {print $0}' <fichier_source> > <fichier_sortie>
# $0 est le code pour la ligne complète.
```